

```

# ----- STACK IMPLEMENTATION -----
class Stack:
    def __init__(self):
        self.s = []

    def push(self, x):
        self.s.append(x)
        print(x, "pushed into Stack")

    def pop(self):
        if not self.s:
            print("Stack Underflow")
        else:
            print("Popped from Stack:", self.s.pop())

    def display(self):
        print("Stack:", self.s)

# ----- QUEUE IMPLEMENTATION -----
class Queue:
    def __init__(self):
        self.q = []

    def enqueue(self, x):
        self.q.append(x)
        print(x, "enqueued into Queue")

    def dequeue(self):
        if not self.q:
            print("Queue Underflow")
        else:
            print("Dequeued from Queue:", self.q.pop(0))

    def display(self):
        print("Queue:", self.q)

# ----- MAIN PROGRAM -----
print("STACK OPERATIONS")
s1 = Stack()
s1.push(10)
s1.push(20)
s1.push(30)
s1.display()
s1.pop()
s1.display()

```

```
print("\nQUEUE OPERATIONS")
q1 = Queue()
q1.enqueue(10)
q1.enqueue(20)
q1.enqueue(30)
q1.display()
q1.dequeue()
q1.display()
```

Output:

STACK OPERATIONS

```
10 pushed into Stack
20 pushed into Stack
30 pushed into Stack
Stack: [10, 20, 30]
Popped from Stack: 30
Stack: [10, 20]
```

QUEUE OPERATIONS

```
10 enqueued into Queue
20 enqueued into Queue
30 enqueued into Queue
Queue: [10, 20, 30]
Dequeued from Queue: 10
Queue: [20, 30]
```